UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/620,078	07/15/2003	Bowen Alpern	YOR920020352US1 (16088)	7092

23389 7590 12/30/2008

SCULLY SCOTT MURPHY & PRESSER, PC
400 GARDEN CITY PLAZA
SUITE 300
GARDEN CITY, NY 11530

EXAMINER	
DAO, THUY CHAN	

ART UNIT	PAPER NUMBER
2192	

MAIL DATE	DELIVERY MODE
12/30/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary

Application No.	Applicant(s)	
10/620,078	ALPERN ET AL.	
Examiner	Art Unit	
Thuy Dao	2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *03 October 2008*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-30* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-30* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *15 July 2003* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

DETAILED ACTION

1. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114.

Applicant's submission filed on October 3, 2008 has been entered.

2. Claims 1-30 have been examined.

Response to Amendments

3. In the instant amendment, claims 1, 20, and 30 have been amended.

4. The objection to the specification and claim 1 is withdrawn in view of Applicant's amendments.

Claim Objections

5. Claims 1, 20, and 30 are objected to because of minor informalities.

Claims 1, 20, and 30:

Claim 1 is the representative claim.

The phrase in the newly added limitation is considered to read as - -*...never call X [[when synchronized]] <u>from within synchronized code</u>*- - as disclosed in the specification, page 9, line 19.

Because other locations recited “software code” (e.g., claim 1, line 1 and its dependent claims), the term “code" (e.g., claim 1, line 3 and other locations) is considered to read as - -<u>software</u> code- -.

Because claim 1, line 14 recites “potential software problems”, the phrase in line 16 is considered to read as - -reporting said <u>potential</u> software problems- -.

Appropriate correction is also requested for independent claims 20 and 30.

Claim 30:

Because a device cannot readable by a machine, the phrase in line 1 is considered to read as - -*A computer program device [[readable by a machine]], tangibly embodying a program of instructions executable by a machine* …- -.

Appropriate correction is requested.

Response to Arguments

6. Applicants' arguments have been considered but are moot in view of the new ground(s) of rejection.

Claim Rejections – 35 USC §101

7. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

8. Claims 20-29 are rejected because the claimed invention is directed to non-statutory subject matter: independent claim 20 directs to "A static analysis framework…", which may comprise only software components such as "means for automatically generating program graphs…", "rule search engine for automatically applying a set of rules…", "means for automatically identifying potential software problems…", and "means for reporting said problems..." (FIG. 4 and related text).

Claim 20 amount(s) to Functional Descriptive Material: "Data Structures" representing descriptive material per se or "Computer Programs" representing computer listings per se.

Data structures not claimed as embodied in computer-readable media are descriptive material per se and are not statutory because they are not capable of causing functional change in the computer. See, e.g., Warmerdam, 33 F.3d at 1361, 31 USPQ2d at 1760 (claim to a data structure per se held nonstatutory). Such claimed data structures do not define any structural and functional interrelationships between the data structure and other claimed aspects of the invention which permit the data structure's functionality to be realized. In contrast, a claimed computer-readable medium encoded with a data structure defines structural and functional interrelationships between the data structure and the computer software and hardware components which permit the data structure's functionality to be realized, and is thus statutory.

Similarly, computer programs claimed as computer listings per se, i.e., the descriptions or expressions of the programs, are not physical "things." They are neither computer components nor statutory processes, as they are not "acts" being performed. Such claimed computer programs do not define any structural and functional interrelationships between the computer program and other claimed elements of a computer which permit the computer program's functionality to be realized. In contrast, a claimed computer-readable medium encoded with a computer program is a computer element which defines structural and functional interrelationships between the computer program and the rest of the computer which permit the computer program's functionality to be realized, and is thus statutory. See Lowry, 32 F.3d at 1583-84, 32 USPQ2d at 1035. Accordingly, it is important to distinguish claims that define descriptive material per se from claims that define statutory inventions. See MPEP 2106.

Dependent claims 21-29 do not cure the deficiencies as noted above, thus, also amount to Functional Descriptive Material: "Data Structures" representing descriptive material per se or “Computer Programs” representing computer listings per se.

Under the principles of compact prosecution, claims 21-29 have been examined as the Examiner anticipates the claims will be amended to obviate these 35 USC § 101 issues. For example, - -A static analysis framework, embedded in a computer, for analyzing software code… - - as disclosed in the specification, page 7, lines 12-13.

Claim Rejections – 35 USC § 103

9. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

10. Claims 1-30 are rejected under 35 U.S.C. 103(a) as being unpatentable over Saxe (art of record, US Patent No. 6,343,376) in view of Koved (art of record, "Access Rights Analysis for Java"), Pinter (art of record, US Patent Publication No. 2002/0129343 A1), and in view of "Checking System Rules Using System-Specific, Programmer-Written Compiler Extensions" to Engler et al. (art made of record, hereafter "Engler").

Claim 1:

Saxe discloses a framework, a computer program device, and *a method for analyzing software code comprising the steps of:*

a) automatically generating program graphs representing runtime characteristics of said code utilizing static analysis techniques (e.g., FIG. 13, col.12: 26-43; col.5: 64 – col.6: 14; col.2: 17-24)*,*

b) automatically applying a set of rules to said program graphs (e.g., FIG. 1, col.5: 64 – col.6: 21; col.2: 46-55; col.3: 63 – col.4: 6)*,*

c) automatically identifying potential software problems from rules set analysis results (e.g., col.6: 16-29; col.6: 61 – col.7: 38)*; and*

d) reporting said soft-ware problems where one or more of best practices violations and coding errors may occur (e.g., FIG. 2, col.7: 1-62).

Saxe does not explicitly discloses *said runtime characteristics including at least adding one or more edges that represent an invocation of a thread.run() which* *results from a call to thread.start(), said runtime characteristics further including at least removing edges from thread.start() to thread.run() when determining which interprocedural nodes are in a thread of execution, said runtime characteristics further including at least adding one or more edges from within an intraprocedural analysis to class constructor based on a rule that specifies when a class constructor must execute.*

However, in an analogous art, Koved further discloses:

said runtime characteristics including at least adding one or more edges that represent an invocation of a thread.run() which results from a call to thread.start() (e.g., page 5, right column, section 5.3 Threads; figure in page 6, adding one edge representing an invocation of "Thread.run()" which results from "Thread.start()"; page 11, Appendix 2, creating the replacement predecessor edge for the Thread.run method),

said runtime characteristics further including at least removing edges from thread.start() to thread.run() when determining which interprocedural nodes are in a thread of execution (e.g., page 4, left column, section 4, the invocation graph with interprocedural is context-sensitive; removing edges/nodes in the invocation graph when two nodes have the same calling context, thus, each node is uniquely identified, i.e., if two allocation sites thread.start() have the same target thread.run() and same calling context, the invocation graph considers them as one unique node by rewriting graph/removing edges),

said runtime characteristics further including at least adding one or more edges from within an intraprocedural analysis to class constructor based on a rule that specifies when a class constructor must execute (e.g., page 3, left column, section 3, each node in the graph represents the intraprocedural analysis; page 4, left column, section 4, the invocation graph includes intraprocedural analysis; page 5, left column, section 5.2 adding edges for FilePermission constructor; page 7, Table 1, when a class constructor must execute, it initializes a class object and the invocation graph creates/includes/adds nodes/edges for class objects and methods).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine Koved's teaching into Saxe's teaching. One would have been motivated to do so to compute the access rights requirements of a program as suggested by Koved (e.g., page 2, left column; page 3, left column, section 3).

Neither Saxe nor Koved explicitly discloses *performing a reachability analysis for at least removing one or more edges to reduce reachability.*

However, in an analogous art, Pinter further discloses *performing a reachability analysis for at least removing one or more edges to reduce reachability* (e.g., FIG. 4, step 46, "Build Reachability Graph to represent references between live variables and objects .generated in allocation statements", page 5, [0062], [0063], and related text).

It would have been obvious to a person of ordinary skill in the art to combine teaching of Pinter into that of Saxe and Koved to better analyze the program flow using intra-procedural reachability analysis as suggested by Pinter (e.g., page 5, [0062]).

Neither Saxe, Koved, nor Pinter explicitly discloses *said set of rules including at least tests for "never call X", "never call X from Y" and "never call X from within synchronized code", said X and Y representing method L signatures.*

However, in an analogous art, Engler further discloses *said set of rules including at least tests for "never call X", "never call X from Y" and "never call X from within synchronized code", said X and Y representing method signatures* (e.g., col.9, Table 1, Sample system rules templates and examples).

It would have been obvious to a person of ordinary skill in the art to combine teaching of Engler into that of Saxe, Koved, and Pinter's teaching to write system-specific compiler extensions that automatically check code for rule violations as suggested by Engler (e.g., col.1 and col.7).

Claim 2:

The rejection of claim 1 is incorporated. Saxe discloses *said rules set represents one or more selected from group comprising: use of best practices and common coding errors, or combinations thereof* (e.g., col.3: 62 – col.4: 16).

Claim 3:

The rejection of claim 1 is incorporated. Saxe discloses *said reporting d) includes presenting results in the context of corresponding source code or object code* (e.g., FIG. 2, col.7: 1-62).

Claim 4:

The rejection of claim 1 is incorporated. Saxe discloses *step b) includes performing rule searches applied to said program graphs* (e.g., col.4: 27-56).

Claim 5:

The rejection of claim 1 is incorporated. Saxe discloses *said software code subject to said static analysis techniques comprises one or more selected from group comprising: object code, source code, a compiler intermediate representation, of said software code, and other program representations, or combinations thereof* (e.g., col.6: 22 – col.7: 27).

Claim 6:

The rejection of claim 3 is incorporated. Saxe discloses *a program graph includes a control analysis graph, said static analysis technique automatically generating said control analysis graphs from said software code* (e.g., col.8: 17-63).

Claim 7:

The rejection of claim 3 is incorporated. Saxe discloses *a program graph includes a data flow analysis graph, said static analysis technique automatically generating said data flow analysis graph from said software code* (e.g., col.9: 15-45).

Claim 8:

The rejection of claim 3 is incorporated. Saxe discloses *a program graph includes an intraprocedural control graph, said static analysis technique automatically generating said intraprocedural control graphs from said software code* (e.g., col.10: 65 – col.11: 33).

Claim 9:

The rejection of claim 3 is incorporated. Saxe discloses *a program graph includes an interprocedural control graphs, said static analysis technique includes automatically generating said interprocedural control graphs from said software code* (e.g., col.12: 26-61).

Claim 10:

The rejection of claim 5 is incorporated. Saxe discloses *said static code analysis further includes automatically identifying classes, fields, methods and class attributes, said set of rules being further applied to said classes and class attributes* (e.g., col.2: 17-24).

Claim 11:

The rejection of claim 5 is incorporated. Saxe discloses *said static code analysis further includes automatically identifying attributes of classes, methods, fields, and aspects of a program' s body* (e.g., col.5: 64 – col.6: 14).

Claim 12:

The rejection of claim 5 is incorporated. Saxe discloses *said step b) further includes the step of: receiving said program graphs and class attributes information and performing a graph rewriting technique* (e.g., col.14: 6-46).

Claim 13:

The rejection of claim 12 is incorporated. Saxe discloses *a result of applying graph rewriting includes generating a run-time characteristics model for said program* (e.g., col.16: 10-58).

It would have been obvious to a person of ordinary skill in the art to combine teaching of Pinter into that of Saxe and Koved to better analyze the program flow using intra-procedural reachability analysis as suggested by Pinter (e.g., page 5, [0062]).

Claim 14:

The rejection of claim 12 is incorporated. Pinter discloses *said step b) further includes the step of receiving said program graphs and attributes information, and performing a reachability analysis* (e.g., FIG. 4, [0061]-[0064]).

Claim 15:

The rejection of claim 14 is incorporated. Pinter discloses *reachability analysis is performed with or without constraints* (e.g., [0055]-[0058]).

Claim 16:

The rejection of claim 14 is incorporated. Pinter discloses *employing a rule search engine to automatically apply a set of rules to said rewrite graph results, reachability analysis results and attributes to identify one or more selected from group of: possible performance errors or problems concerning correctness, security, privacy and maintainability of said software code* (e.g., [0017]-[0022]).

It would have been obvious to a person of ordinary skill in the art to combine teaching of Pinter into that of Saxe and Koved to better analyze the program flow using intra-procedural reachability analysis as suggested by Pinter (e.g., page 5, [0062]).

Claim 17:

The rejection of claim 14 is incorporated. Saxe discloses *said rewrite graph technique includes traversing a program graph to locate nodes containing attributes of interest and to locate edges to add or remove from said program graph* (e.g., col.17: 15 – col.18: 37).

Claim 18:

The rejection of claim 17 is incorporated. Pinter discloses *said reachability analysis includes traversing the program graphs and adding or removing edges to extend or reduce reachability, respectively* (e.g., [0029]-[0035]).

Claim 19:

The rejection of claim 18 is incorporated. Saxe discloses *a rule is applied to determine whether a node representing a particular method is reachable by traversing said graph from a particular head node, said head node being user selectable* (e.g., col.21: 10 – col.22: 6).

Claim 20:

Claim 20 recites the same limitations as those of claim 1, wherein all claimed limitations have been addressed and/or set forth above. Therefore, as the references teach all of the limitations of the above claim, they also teach all of the limitations of claim 1.

Claim 21:

The rejection of claim 20 is incorporated. Saxe discloses *said rules set represents one or more selected from group comprising: use of best practices and common coding errors, or combinations thereof* (e.g., col.3: 63 – col.4: 6).

Claim 22:

The rejection of claim 20 is incorporated. Saxe discloses *said software code comprises scalable componentized applications according to a software development platform* (e.g., col.6: 16-29).

Claim 23:

The rejection of claim 20 is incorporated. Saxe discloses *said program graphs include one or more selected from group comprising: a control analysis t graph, a data flow analysis graph, an intraprocedural control flow graph and an interprocedural control flow graph, said static analysis technique automatically generating a respective one of said control analysis graph, data flow analysis graph, intraprocedural control flow graph and interprocedural control flow graph from said software code* (e.g., col.7: 1-62).

Claim 24:

The rejection of claim 23 is incorporated. Saxe discloses *means for automatically identifying classes, fields, methods and class attributes, said set of rules being further applied to said classes and class attributes* (e.g., col.4: 27-56).

Claim 25:

The rejection of claim 23 is incorporated. Saxe discloses *said static code analysis further includes automatically identifying attributes of classes, methods, fields, and aspects of a program's body* (e.g., col.6: 22 – col.7: 27).

Claim 26:

The rejection of claim 20 is incorporated. Saxe discloses *said means for automatically generating program graphs includes means for performing graph rewriting* (e.g., col.8: 17-63).

Claim 27:

The rejection of claim 26 is incorporated. Saxe discloses *results of said graph rewriting include a run-time characteristics model for said program* (e.g., col.14: 6-46).

Claim 28:

The rejection of claim 26 is incorporated. Saxe discloses *said means for automatically generating program graphs includes: means for performing a reachability analysis, said reachability analysis being performed with or without constraints* (e.g., col.12: 26-61).

Claim 29:

The rejection of claim 28 is incorporated. Saxe discloses *said rule search engine automatically applies a set of rules to said rewrite graph results, reachability analysis results and attributes to identify one or more of: possible performance errors or problems concerning correctness, security and privacy of said software code* (e.g., col.10: 65 – col.11: 33).

Claim 30:

Claim 30 recites the same limitations as those of claim 1, wherein all claimed limitations have been addressed and/or set forth above. Therefore, as the references teach all of the limitations of the above claim, they also teach all of the limitations of claim 1.

Conclusion

11. Any inquiry concerning this communication should be directed to examiner Thuy Dao (Twee), whose telephone/fax numbers are (571) 272 8570 and (571) 273 8570, respectively. The examiner can normally be reached on every Tuesday, Thursday, and Friday from 6:00AM to 6:00PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam, can be reached at (571) 272 3695.

The fax phone number for the organization where this application or proceeding is assigned is (571) 273 8300.

Any inquiry of a general nature of relating to the status of this application or proceeding should be directed to the TC 2100 Group receptionist whose telephone number is (571) 272 2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Thuy Dao/
Examiner, Art Unit 2192

/Tuan Q. Dam/
Supervisory Patent Examiner, Art Unit 2192